Intern. J. Computer Math. 1981, Vol. 9, pp. 353–361
0020-7160/81/0904-0353 $06.50/0

Printed in Great Britain

On a Family of Multipoint Methods for Non-linear Equations

BENY NETA†

Northern Illinois University, Department of Mathematical Sciences, DeKalb, Illinois 60115

(Received July 1981)

A new one-parameter family of methods for finding simple zeros of non-linear functions is developed. Each member of the family requires four evaluations of the given function and only one evaluation of the derivative per step. The order of the method is 16.

KEY WORDS: Non-linear equations, order of convergence, Newton's method, zero, root, iteration.

C.R. CATEGORY: 5.1.

1. INTRODUCTION

Newton's method for computing a simple zero ζ of a non-linear equation $f(x)=0$ has been modified in a number of ways. For example, Ostrowski [14] discusses a third-order method that evaluates the function f at every substep but only requires the derivative f' at every other substep. He also introduced a fourth-order scheme that uses the same information. King [9] has shown that there is a family of such methods. Traub [15] introduced a third-order method which requires one function and two derivative evaluation per step. Jarratt [6] developed a fourth-order method which uses the same information. King [8] developed a fifth-order scheme that requires two evaluations of f and f'. Werner [16] introduced a method of order $1+\sqrt{2}$ that requires one evaluation of f and f'. And recently, the author [13] developed a family of sixth-order methods that requires 3 evaluations of f and one of f'.

†Now at Department of Mathematics, Texas Tech. University, Lubbock, Tx 79409.

Here we construct a method of order 16. An iteration consists of one Newton substep followed by a substep of "modified" Newton (i.e., using the derivative of f at the first substep instead of the current one) and then two substeps of inverse interpolation.

Let us recall the definition of order (see e.g. [15]).

DEFINITION 1 Let $x_1, x_2, \ldots, x_i$ be a sequence converging to ζ. Let

$$\varepsilon_i = x_i - \zeta. \tag{1}$$

If there exists a real number p and a nonzero constant C such that

$$\frac{|\varepsilon_{i+1}|}{|\varepsilon_i|^p} \rightarrow C \tag{2}$$

then p is called the *order* of the sequence.

There are two other concepts related to order: one measures the information used and the other measures the efficiency.

DEFINITION 2 The *informational usage* d of a scheme is defined as the number of *new* pieces of information required per iteration.

DEFINITION 3 The *informational efficiency* EFF of a scheme is defined as the order p divided by the informational usage d.

$$\mathrm{EFF} = \frac{p}{d} \tag{3}$$

DEFINITION 4 The *efficiency index* *EFF is defined by

$$^{*}\mathrm{EFF} = p^{1/d} \tag{4}$$

where p and d are as in definition 3. (This term was introduced by Ostrowski [14]).

In the following section we develop the scheme. Section 3 will be devoted to compare the efficiency of all known methods. Section 4 contains a small numerical example.

2. DEVELOPMENT OF THE METHOD

Let

$$
\begin{aligned}
w_n &= x_n - \frac{f(x_n)}{f'(x_n)} \\
z_n &= w_n - \frac{f(w_n)}{f'(x_n)} \frac{f(x_n) + Af(w_n)}{f(x_n) + (A-2)f(w_n)} \qquad (5) \\
t_n &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n) - f(w_n)}{f(x_n) - 3f(w_n)}.
\end{aligned}
$$

If we let $x_{n+1} = t_n$ we obtain the sixth-order family in [13]. Suppose we compute x_{n+1} by inverse interpolation. Let

$$
\begin{aligned}
R(f(x)) = a + b(f(x) - f(x_n)) + c(f(x) - f(x_n))^2 \\
+ d(f(x) - f(x_n))^3 + e(f(x) - f(x_n))^4 \qquad (6)
\end{aligned}
$$

be a polynomial of degree four satisfying

$$
\begin{aligned}
x_n &= R(f(x_n)) \\
\frac{1}{f'(x_n)} &= R'(f(x_n)) \\
w_n &= R(f(w_n)) \qquad (7) \\
z_n &= R(f(z_n)) \\
t_n &= R(f(t_n))
\end{aligned}
$$

It is easy to see from the first two equations of (7) that

$$
\begin{aligned}
a &= x_n \qquad (8) \\
b &= \frac{1}{f'(x_n)}
\end{aligned}
$$

Thus, if we use the notations,

$$\begin{aligned} \delta &= \delta_n - x_n \\ F_\delta &= f(\delta_n) - f(x_n) \qquad (9) \\ \phi_\delta &= \frac{\delta}{F_\delta^2} - \frac{1}{F_\delta f'(x_n)}, \quad \text{for} \quad \delta = w, z, t \end{aligned}$$

then the last three equations of (7) will give

$$\begin{aligned} c + dF_w + eF_w^2 &= \phi_w \\ c + dF_z + eF_z^2 &= \phi_z \qquad (10) \\ c + dF_t + eF_t^2 &= \phi_t. \end{aligned}$$

Solving these equations we have:

$$\begin{aligned} e &= \frac{\dfrac{\varphi_t - \phi_z}{F_t - F_z} - \dfrac{\phi_w - \phi_z}{F_w - F_z}}{F_t - F_w} \\ d &= \frac{\phi_t - \phi_z}{F_t - F_z} - e(F_t + F_z) \qquad (11) \\ c &= \phi_t - dF_t - eF_t^2 . \end{aligned}$$

Once the coefficients were computed, then

$$x_{n+1} = R(0) = x_n - \frac{f(x_n)}{f'(x_n)} + cf^2(x_n) - df^3(x_n) + ef^4(x_n). \qquad (12)$$

We would like to show that the scheme (5), (12) is of order $p = 14$. To this end, we use a result of Traub [15].

THEOREM (TRAUB) *Let x_i, $x_{i-1}, \ldots, x_{i-n}$ be $n+1$ approximations to a zero ζ of f. Let $Q_{n,\gamma}$ be the interpolatory polynomial at y_i, $y_{i-1}, \ldots, y_{i-n}$ in the sense of*

$$\begin{aligned} Q_{n,\gamma}^{(k_j)}(y_{i-j}) = \mathscr{F}^{(k_j)}(y_{i-j}) \quad &\text{for} \quad j = 0, 1, 2, \ldots, n \\ k_j = 0, 1, \ldots, \gamma_{j-1}, \quad &\gamma_j \geqq 1 \qquad (13) \end{aligned}$$

where $\mathscr{F}$ is the inverse of f.

Define a new approximation to ζ by

$$x_{i+1}=Q_{n,\gamma}(0). \tag{14}$$

and let

$$e_i=x_i-\zeta, \tag{15}$$

then

$$e_{i+1}=M_i\prod_{j=0}^{n}e_{i-j}^{\gamma_j} \tag{16}$$

for suitable constants M_i.

In our case

$$n=3$$

$$\gamma_0=\gamma_1=\gamma_2=1$$

$$\gamma_3=2.$$

Note that

$$e_i\sim e_{i-3}^{8}, \tag{17}$$

$$e_{i-1}\sim e_{i-3}^{4} \tag{18}$$

(The scheme constructed from the first two substeps of (5) is of order 4, see [9]), and

$$e_{i-2}\sim e_{i-3}^{2}. \tag{19}$$

Substituting these in (16) yields

$$e_{i+1}\sim e_{i-3}^{6}e_{i-3}^{4}e_{i-3}^{2}e_{i-3}^{2}=e_{i-3}^{14}. \tag{20}$$

Thus the order is $p=14$.

One can improve the order of this scheme just by replacing the third substep of (5) by a step similar to the fourth one. Let

$$T(f(x)) = \alpha + \beta(f(x) - f(x_n)) + \gamma(f(x) - f(x_n))^2 + \delta(f(x) - f(x_n))^3 \tag{21}$$

be a cubic polynomial satisfying

$$\begin{aligned} x_n &= T(f(x_n)) \\ \frac{1}{f'(x_n)} &= T'(f(x_n)) \\ w_n &= T(f(w_n)) \\ z_n &= T(f(z_n)) \end{aligned} \tag{22}$$

Clearly,

$$\begin{aligned} \alpha &= x_n \\ \beta &= \frac{1}{f'(x_n)}. \end{aligned} \tag{23}$$

Using the notations of (9), the last two equations of (22) will be

$$\begin{aligned} \gamma + \delta F_w &= \phi_w \\ \gamma + \delta F_z &= \phi_z \end{aligned}. \tag{24}$$

The coefficients γ, δ are given by

$$\begin{aligned} \delta &= \frac{\phi_w - \phi_z}{F_w - F_z} \\ \gamma &= \phi_w - \delta F_w \end{aligned} \tag{25}$$

Once the coefficients were computed, then

$$t_n = T(0) = x_n - \frac{f(x_n)}{f'(x_n)} + \gamma f^2(x_n) - \delta f^3(x_n). \tag{26}$$

In order to obtain the order of the scheme composed of the first two substeps of (5), (26) and (12), one uses (16). To this end, we need e_i which is the error at the third substep. It can be computed from (16)

$$e_i = M_{i-1} e_{i-1} e_{i-2} e_{i-3}^2. \tag{27}$$

Combining (16), (27) with (18)-(19) one obtains

$$e_{i+1} \sim e_{i-3}^{16}$$

Thus, the order $p = 16$.

3. EFFICIENCY OF ITERATIVE METHODS

In Section 1 we stated two definitions for efficiency. In the following we compare the efficiency of all methods mentioned in Section 1 with our new method. This is done in two tables. In Table I the methods listed in decreasing order of informational efficiency. This shows that our new method is the most efficient one. In Table II we list the methods in decreasing order of efficiency index. Our new method comes second, next to Muller's method.

TABLE I

Method	*Reference*	*Order*	*Informational Usage*	*EFF*
Neta	This paper	16	5	3.2
Neta	This paper	14	5	2.8
Muller	[11]	1.839	1	1.839
Pegasus	[5]	7.275	4	1.819
Anderson and Björck	[1]	8	4	2
		5	3	1.667
Secant	[3]	1.618	1	1.618
Improved Pegasus	[10]	5	3	1.667
		3	2	1.5
Neta	[13]	6	4	1.5
Jarratt	[6]	4	3	1·333
Ostrowski	[14]	4	3	1.333
King	[9]	4	3	1.333
King	[8]	5	4	1.25
Murakami	[12]	5	4	1.25
Werner	[16]	$1+\sqrt{2}$	2	1.207
Ostrowski	[14]	3	3	1
Traub	[15]	3	3	1
Snyder	[4]	3	3	1
Newton	[7]	2	2	1
Steffensen	[7]	2	2	1

TABLE II

Method	*Reference*	*Order*	*Informational Usage*	**EFF*
Muller	[11]	1.839	1	1.839
Neta	This paper	16	5	1.741
Improved Pegasus	[10]	3	2	1.732
		5	3	1.710
Anderson and Björck	[1]	5	3	1.710
		8	4	1.682
Neta	This paper	14	5	1.695
Pegasus	[5]	7.275	4	1.642
Secant	[3]	1.618	1	1.618
Ostrowski	[14]	4	3	1.587
King	[9]	4	3	1.587
Jarratt	[6]	4	3	1.587
Neta	[13]	6	4	1.565
Werner	[16]	$1+\sqrt{2}$	2	1.554
King	[8]	5	4	1.495
Murakami	[12]	5	4	1.495
Ostrowski	[14]	3	3	1.442
Traub	[15]	3	3	1.442
Snyder	[4]	3	3	1.442
Newton	[7]	2	2	1.414
Steffensen	[7]	2	2	1.414

4. NUMERICAL EXAMPLE

Let $f(x)=x^3+\ln(1+x)$ where ln denotes the logarithm to the natural base. Hence $\zeta=0$. Starting at $x_0=0.1$, 0.2, 0.3, 0.4 and 0.5 we compute x_1 by our lower order $(p=14)$ algorithm. Calculations were done in double precision arithmetic on IBM 370/148 computer.

Results are summarized in the following table. The parameter A that appears in the algorithm was chosen $A=2$.

TABLE III

x_0	x_1	x_0^{14}
0.1	$0.3904.10^{-15}$	10^{-14}
0.2	$-0.9487.10^{-15}$	$0.1638.10^{-9}$
0.3	$-0.5323.10^{-8}$	$0.4783.10^{-7}$
0.4	$-0.3075.10^{-6}$	$0.2684.10^{-5}$
0.5	$-0.2899.10^{-6}$	$0.6104.10^{-4}$

Note that x_1 is closer to ζ than x_0^{14}. In order to reliably determine the order one would have to use higher precision, see e.g. [2].

References

[1] N. Anderson and Å. Björck, A new high order method of Regula Falsi type for computing a root of an equation, *BIT* **13** (1973), 253–264.

[2] P. Brent, A FORTRAN multiple-precision arithmetic package, *ACM Trans. on Math. Software* **4** (1978), 56–70.

[3] S. D. Conte and C. deBoor, Elementary Numerical Analysis: An Algorithmic Approach, Second Edition, McGraw Hill Book Company, NY, 1972.

[4] M. Dowell and P. Jarratt, A modified Regula Falsi method for computing the root of an equation, *BIT* **11** (1971), 168–174.

[5] M. Dowell and P. Jarratt, The "Pegasus" method for computing the root of an equation, *BIT* **12** (1972), 503–508.

[6] P. Jarratt, Some efficient fourth-order multipoint methods for solving equations, *BIT* **9** (1969), 119–124.

[7] L. W. Johnson and R. D. Riess, Numerical Analysis, Addison–Wesley Pub. Co., Mass., 1977.

[8] R. F. King, A fifth-order family of modified Newton methods, *BIT* **11** (1971), 409–412.

[9] R. F. King, A family of fourth-order methods for nonlinear equations, *SIAM J. Numer. Anal.* **10** (1973), 876–879.

[10] R. F. King, An improved Pegasus for root finding, *BIT* **13** (1973), 423–427.

[11] D. E. Muller, A method for solving algebraic equations using an automatic computer, *Math. Comp.* **10**, (1956), 208–215.

[12] T. Murakami, Some fifth-order multipoint iterative formulae for solving equations, *J. of Information Processing* **1** (1978), 138–139.

[13] B. Neta, A sixth-order family of methods for nonlinear equations, *Intern. J. Computer Math.* **7** (1979), 157–161.

[14] A. M. Ostrowski, Solution of Equations and Systems of Equations, Academic Press, NY 1960.

[15] J. F. Traub, Iterative Methods for the Solution of Equations, Prentice–Hall, Englewood-Cliffs, NJ, 1964.

[16] W. Werner, Uber ein verfhren der ordnung $1+\sqrt{2}$ zur nullstellenbestimmung, *Numer. Math.* **32** (1979), 333–342.